

## CLAIMS

1. A processor-readable medium comprising processor-executable instructions for:  
parsing an input file to recognize a file format of the input file;  
checking contents of the input file, according to the recognized file format  
if available, to determine whether executable code exists within the  
input file; and  
sending a status in response to results of said checking.
2. The processor-readable medium as recited in claim 1, wherein sending a status comprises further instructions for:  
sending a file-has-no-code status when the file format of the input file was  
recognized and no executable code was found; and  
sending a file-has-code status when executable code was found.
3. The processor-readable medium as recited in claim 1, additionally comprising further instructions for sending a don't-know status when the file format of the input file was not recognized.
4. The processor-readable medium as recited in claim 1, wherein parsing the input file comprises further instructions for parsing the input file repeatedly with a plurality of component parsers contained within an extensible parser.

5. The processor-readable medium as recited in claim 4, additionally comprising further instructions for sending a don't-know status when each of the plurality of component parsers contained within the extensible parser fails to recognize sections of the input file.
6. The processor-readable medium as recited in claim 1, wherein sending the status comprises further instructions for sending the status to an email program.
7. The processor-readable medium as recited in claim 1, wherein sending the status comprises further instructions for sending the status to an instant messaging program.
8. The processor-readable medium as recited in claim 1, wherein sending the status comprises further instructions for sending the status to an internet browsing program.
9. The processor-readable medium as recited in claim 1, wherein checking contents of the input file comprises further instructions for operating a component parser portion of an extensible parser to recognize a specific data file format.

10. The processor-readable medium as recited in claim 1, wherein parsing the input file comprises instructions for operating a parser configured to recognize a plurality of file formats.
11. The processor-readable medium as recited in claim 1, wherein checking contents of the input file comprises further instructions for operation of a compound parser including a plurality of component parsers, wherein each of the plurality of component parsers is configured for recognition of a specific file format.
12. The processor-readable medium as recited in claim 11, additionally comprising further instructions for continuing to parse the input file with all remaining component parsers after at least one component parser recognizes the file format of the input file.
13. The processor-readable medium as recited in claim 11, additionally comprising further instructions for parsing the input file until a component parser recognizes the file format of the input file.

14. A method of detecting code-free files, comprising:  
parsing an input file with a compound parser configured to include a plurality of component parsers, wherein each component parser is configured to recognize a specific data file format;  
analyzing contents of the input file according to the recognized specific file format, where available, to determine if the input file contains executable code; and  
sending a status in response to results of said analyzing.
15. The method as recited in claim 14, additionally comprising:  
sending a file-has-no-code status when the file format of the input file was recognized and no executable code was found; and  
sending a file-has-code status when executable code was found.
16. The method as recited in claim 14, additionally comprising sending a don't-know status when a file format of the input file was not recognized.
17. The method as recited in claim 14, additionally comprising sending the status to an email program.
18. The method as recited in claim 14, additionally comprising sending the status to an instant messaging program.
19. The method as recited in claim 14, additionally comprising sending the status to an internet browsing program.

20. The method as recited in claim 14, wherein parsing the input file comprises parsing the input file with each of the plurality of component parsers within the compound parser.
21. An apparatus for detecting code-free files, comprising:  
a compound parser configured to repeatedly parse an input file, wherein each component parser within the compound parser is configured to recognize executable code within a specific file format selected from among a group of data file formats; and  
a controller to examine success of each of the component parsers to recognize the specific file format for which it was configured to recognize and to find executable code within the input file, wherein the controller is configured to send a status in response to results of said checking.
22. The apparatus as recited in claim 21, additionally configured so that if none of the component parsers recognizes the file format of the input file, a don't-know status is sent.
23. The apparatus as recited in claim 21, wherein the apparatus for detecting code-free files is additionally configured to send the status to an email program.

24. The apparatus as recited in claim 21, wherein the apparatus for detecting code-free files is additionally configured to send the status to an instant messaging program.
25. The apparatus as recited in claim 21, wherein the apparatus for detecting code-free files is additionally configured to send the status to an internet browsing program.
26. The apparatus as recited in claim 21, additionally configured to send the status to:
  - a firewall;
  - a host intrusion detector; or
  - a host vulnerability assessor.
27. The apparatus as recited in claim 21, additionally configured to send the status to a program selected from a group of programs, comprising:
  - a backup program;
  - a CD/DVD burning program; and
  - a P2P file-sharing program.
28. The apparatus as recited in claim 21, wherein each of the component parsers is configured to recognize one of a plurality of data file formats.

29. The apparatus as recited in claim 21, wherein the compound parser is configured to allow extension by addition of a new component parser to the compound parser, wherein the new component parser recognizes a further file format and recognizes executable code within the further file format.